

BBS Payment Solutions

PA-DSS Implementation Guide

Information for Integrators

BBS Denmark A/S

Author: Torben Ellgaard

Revision: B

Date: 2010-07-02

Product Scope

This document covers solutions and products where the Payment Application is equivalent of applications delivered by Sagem Denmark A/S in the past.

These applications are named:

- Sagem PS (PSAM solution)
- Sagem GPA (other solutions).

About this Document

1.1 Scope and Validity

This document provides information for integrators of the BBS Denmark Payment Application. The guidelines given in this document must be observed in order not to hinder PCI DSS compliance of the completed implementation.

Document release date: 2010-04-16

Covering the following versions of PCI requirements:

PCI DSS version 1.2, launched October 1, 2008

PA-DSS version 1.2, launched October 1, 2008

The newest release of PCI requirements can be found at:

<https://www.pcisecuritystandards.org>

This document is applicable to the following Payment Application versions listed as PA-DSS compliant on the BBS Denmark website:

<http://www.bbs.dk/pa-dss>

This document will be reviewed at least once a year or when new versions of the requirements or the application are released.

Please contact BBS Denmark if this document appears to be out of date or doesn't cover the version you are using.

1.2 Abbreviations

API	Application Programming Interface
Flexi	Indoor attended payment terminal offered by Sagem Denmark
GPA	Generic Payment Application – framework used by Sagem Denmark to create Payment Applications for various acquirers (except for PBS)
PA-QSA	Payment Application QSA – authorised by PCI to perform PA-DSS evaluations of Payment Applications
PAN	Primary Account Number – the cardholders account number with the card issuer PCI Payment Card Industry – Organisation founded by Visa, MasterCard, JCB, Discover Card and American Express
PBS	
PCI DSS	Data Security Standard, PCI requirements for protecting cardholder data
PCI PA-DSS	Payment Application Data Security Standard, PCI requirements for the Payment Application related to PCI DSS
PCI SSC	Security Standards Council – comity in PCI managing the requirements
PED	PIN Entry Device – the secure device used for PIN entry and encryption
PIN	Personal Identification Number

PS	Payment Solution – the Payment Application used with the PBS PSAM concept (PBS)
PSAM	Payment Security Application Module – a small SIM-like chipcard used by the Danish acquirer PBS
PTS	PIN Terminal Security – new PCI term for PED
QSA	Qualified Security Assessor – authorised by PCI to perform PCI DSS evaluations
SSH	Secure Shell – protocol to authenticated and encrypted login
UCM	Universal Controller Module – a HW platform for the Payment Application in unattended solutions

1.3 References

- [1] Payment Card Industry (PCI), Data Security Standard Requirements and Security Assessment Procedures Version 1.2, October 2008
- [2] Payment Card Industry (PCI), Payment Application Data Security Standard Requirements and Security Assessment Procedures Version 1.2, October 2008
- [3] Installation “readme” for Danish Payment Application (PS)
File: Install_eng.txt
- [4] Installation of GPA based Payment Application
File: GPA-Demo Installation and Usage verB.pdf
- [5] Programmers Guide for Integrators on how to use the Java Payment API
File: Programmers guide for integrators.pdf
- [6] JavaDoc with specification of methods in the Payment API (html-document)
File: interfaces.zip
- [7] COM Bridge for integration in Windows environment
File: Sagem ComBridge Specification SP-414-0179 Rev A 090810.pdf
- [8] Serial Interface Layer – Protocol description
File: Serial interface34_External_.pdf

1.4 Revision Log

Date	Changed by	Description of change	Revision
2010-07-02	TE	Correction of PCI DSS level values 0 means no full PAN	B
2010-04-16	TE	Release of document after PA-DSS approval of the applications	A
2009-11-06	TE	Document name changed to PA-DSS Implementation Guide Appendix on wipe tool add	GG

		Related correction of paragraph 4.3.1 Added Chapter 7 on Key Management and key replacement.	
2009-10-29	TE	Submitted to QSA	GF
2009-10-15	TE	Internal review	GE
2009-10-08	TE	Internal review	GD
2009-07-14	TE	Internal review	GC
2009-07-10	TE	Second review	GB
2009-05-20	TE	Draft release for review	GA

Disclaimer

The information contained in this document is correct to the best of our knowledge.

Information given in this document has been evaluated by a PA-QSA as part of the PA-DSS process.

Statements on processes outside the payment application are given as guidelines, it is recommended that they are verified by a QSA when implementing solutions to achieve PCI DSS compliance.

1.5 Table of Contents

1	About this Document.....	2
1.1	Scope and Validity	2
1.2	Abbreviations	2
1.3	References.....	3
1.4	Revision Log	3
1.5	Table of Contents.....	5
2	Introduction	6
3	Impact of PCI DSS Requirements.....	7
3.1	PA-DSS Compliant Payment Application	7
3.1.1	Storage of PAN.....	7
3.1.2	Storage of sensitive authentication data	8
3.1.3	Access to cardholder data	8
3.1.4	Transmission of cardholder data.....	8
3.2	Procedures in relation to PA-DSS.....	8
3.3	PA-DSS Implementation Guide (this document)	8
4	Utilising the Payment Application	9
4.1	Platform for the Payment Application.....	9
4.2	Integration of Payment Application	11
4.3	Installation of Payment Application	11
4.3.1	Clean-up from previous application (full-integration).....	11
4.4	Administrator Access (Configuration).....	12
4.4.1	Full Integration	12
4.4.2	Embedded Integration (first generation platform).....	12
4.4.3	Embedded Integration (second generation platform).....	12
4.5	Verification of PCI DSS Settings	15
4.6	Restrictions on storing Cardholder Data	15
4.7	Network Connection.....	15
4.7.1	Wireless Networks	16
4.8	SW Updates.....	16
4.8.1	Updating full integration deployments	16
4.8.2	Updating embedded deployments	17
4.9	Trouble-shooting.....	17
4.9.1	Debug-Logging	17
4.9.2	Support Procedures.....	17
5	Configuration File Settings.....	17
5.1	General Security Settings	17
5.1.1	Selection of Crypto Driver	18
5.1.2	receipt.printFullPan.....	18
5.2	API Settings for PAN Formats.....	19
5.2.1	Offered PAN formats.....	19
5.2.2	PAN channels in API.....	19

2 Introduction

One of the recent challenges to merchants and payment solution integrators is the requirements of PCI DSS.

The PCI DSS requirements address the dilemma that the card data required in legitimate processing and handling of magnetic stripe payment transactions are sufficient for performing the transaction. This makes magnetic stripe card data a valuable that has to be protected. Unlike chip cards, the sensitive data are accessible for electronic distribution and storage and the card itself is not needed to make a fraud transaction.

The Payment Application will always handle card data in the course of a transaction. If this application is provided by a third party, the integrator/merchant may not have knowledge of the details of its operation.

By obtaining a PA-DSS approval, the application vendor demonstrates that the Payment Application itself doesn't obstruct PCI DSS compliance.

This document has three main chapters covering:

Impact of PCI DSS Requirements

This is an overview of features offered by the PA-DSS compliant Payment Application

Utilising the Payment Application

Instructions for use of the Payment Application in a PCI DSS compliant manner

Details on the Configuration

Describes the individual parameters that can be configured to assist in achieving PCI DSS compliance.

Appendix A describes the procedure required to stay compliant with PCI DSS requirements when configuring "first generation" products.

Appendix B describes how to use the hash-function to generate a unique hash value from the PAN.

Appendix C describes secure wipe of data left by previous versions of the application

3 Impact of PCI DSS Requirements

The impact of the PCI DSS requirements will depend on the specific implementation and on the agreements between the merchant and the acquirer.

As a general rule a PA-DSS compliant Payment Application is a step towards PCI DSS compliance, but it doesn't ensure compliance; neither does it relieve the merchant of the responsibility to demonstrate compliance.

The acquirers are responsible for ensuring PCI DSS compliance with the merchants they service. It is important that the specific requirements from the acquirer are checked to find the right method for demonstrating compliance.

3.1 PA-DSS Compliant Payment Application

The full list of requirements for the Payment Application can be found in the PCI PA-DSS document [1]. Most of these requirements have been best practise in payment applications for a long time and they have been part of the payment schemes defined by acquirers.

As a result the PA-DSS compliant application differs very little from the previous versions. PA-DSS compliance is only available in new versions of the Payment Application (see 1.1). Merchants/integrators that require PCI DSS compliance are advised to migrate to PA-DSS compliant versions.

Three features have been implemented in the PA-DSS compliant application:

- PAN data stored temporarily are stored in encrypted form
- The PAN format delivered by the application in various parts of the Payment API is now configurable (full, truncated, omitted etc).
- PAN data has been removed from trace log files used for debugging

The first feature is purely internal to the Payment Application. Details of the Payment Application features are given in the paragraphs below.

The second feature is not a strict PA-DSS requirement, but it may assist the integrator in demonstrating PCI DSS compliance. The new API configurations are described in detail in paragraph 5.2 of this document.

The third modification ensures that debugging with the "trace.log" file is still available without compromising PCI DSS requirements. Naturally part of the information will be missing, but the "trace.log" can still be recorded as described in paragraph 4.9.1.

3.1.1 Storage of PAN

The Payment Application stores the PAN in encrypted form and keeps it until an acknowledgement of the Financial Advice has been received from the host. When the transaction has been confirmed the encrypted PAN is deleted.

Depending on the configuration the full PAN may be decrypted for inclusion in the Financial Advice sent to the host or in the journal data provided to the merchant application.

3.1.2 Storage of sensitive authentication data

The Payment Application does not store any sensitive authentication data at any time.

3.1.3 Access to cardholder data

The Payment Application does not provide a user login with access to neither PAN nor sensitive authentication data. (PA-DSS 3.1).

3.1.4 Transmission of cardholder data

Cardholder data transmitted to the acquirer host is protected through encryption. The encryption is defined by the acquirer and performed either in the secure PED or by the PSAM (PBS). The encryption provided by the application fulfils the requirements of PCI DSS to allow transmission over public networks (PA-DSS 12.1).

The Payment Application doesn't support any transmission of PAN or sensitive authentication data through end-user messaging technologies (PA-DSS 12.2).

3.2 Procedures in relation to PA-DSS

Apart from the specific features of the application listed above, the PA-DSS requirements also address vendor procedures of related to development, testing, deployment and support. Some of these procedures may affect the integrator of the application.

Details are provided in this document on how to verify the authenticity of payment application and how to handle trouble-shooting issues. See paragraphs 4.8.1 and 4.9.2.

3.3 PA-DSS Implementation Guide (this document)

In addition to the features just described and the procedures, PA-DSS also requires that the vendor provides a PA-DSS Implementation Guide.

The implementation guide (this document) provides instructions on how to integrate, operate, troubleshoot and update the application in a PCI DSS compliant manner.

Some paragraphs of the PA-DSS requirements demands that specific issues are covered in this guide. References to these paragraphs are given in brackets to facilitate the review of the document. The references may be used to seek further information in the PA-DSS requirements.

On some issues it is required that PCI DSS requirements are quoted in the document. The quotes are presented in a different font as illustrated by this paragraph.

4 Utilising the Payment Application

This chapter describes how the Payment Application should be used in order not to hinder PCI DSS compliance of the complete payment solution.

4.1 Platform for the Payment Application

The Payment Application will operate on platforms that support a suitable Java Runtime Environment (see [3] – [4]). When used in a full integration scenario the HW platform and the Java environment is provided by the integrator.

For embedded solutions BBS Denmark delivers the HW platform fully configured with operating system and Java Runtime Environment. Several versions of the embedded platform exists:

- The older version of Flexi and all Flexi Solo stand-alone terminals
- New PCI PTS approved Flexi and all unattended UCM solutions
- ML30, stand-alone terminals Flexi Mobile and Flexi Combi

The table on the following page provides an overview of the platforms for the Payment Application. Details on the columns are given in the paragraphs that follow:

- Administrator Access in paragraph 4.4
- SW Updates in paragraph 4.8

PS (with PSAM)					
Ref.	Platform	Characteristics	Administrator Access	SW Update	PCI DSS Requirements
PS1	Full integration (integrator)	Payment application is installed on platform provided by integrator. Terminals for integration can be the original Flexi, the PCI PTS approved Flexi or PIN pads for unattended use.	Handled by integrator	Handled by integrator	Restricted access with username and password
PS2	Embedded integration (JBed, Flexi 1)	Uses first generation Flexi terminal without backlight in keys.	No remote login (HTTP removed)	Connects to ftp-server	Configuration only through the procedure described in Appendix A
PS3	Embedded integration (Linux, Flexi 2, UCM)	Uses second generation Flexi terminal with backlight in keys or UCM for unattended solutions.	Console and SSH	Connects to ftp-server	Restricted access through SSH with configured username and password
PS4	Embedded integration (ML30, Posix)	Uses first or second generation Flexi terminal.	No		
PS5	Stand-alone Flexi Solo (JBed)		No remote login (HTTP removed)	Connects to ftp-server	Configuration only through the procedure described in Appendix A
PS6	Stand-alone Flexi Mobil (930, Posix)		No		None, configuration isn't possible
GPA					
Ref.	Platform	Characteristics	Administrator Access	SW Update	PCI DSS Requirements
GPA10	Full integration (integrator)	GPA installed on platform provided by integrator. Terminals for integration can be the original Flexi, the PCI PTS approved Flexi or PIN pads for unattended use.	Handled by integrator	Handled by integrator	

4.2 Integration of Payment Application

Details of various interfaces for integration are given in specific documents ([3] – [9]).

In the context of this document it is sufficient to distinguish between full integration in which the integrator provides the platform for the application and embedded integration where the payment application is installed and delivered in the terminal by Sagem Denmark. In the following it will be indicated if a description is specific to the form of integration.

The issue to consider when integrating the PA-DSS compliant Payment Application is how to configure the full PAN presence on the integration interface. The Payment Application includes a configuration file for defining these formats of the PAN in different parts of the API. Paragraph 5.2 describes how these settings are used.

Default setting will block PAN information and deliver truncated PAN information.

If the payment application is configured to deliver full PAN information, the integrator/merchant must treat this information according to the PCI DSS rules.

4.3 Installation of Payment Application

Instructions for installation in full integration are given elsewhere ([3] and [4]).

Embedded terminals are delivered pre-installed with operating system, Java Runtime Environment and Payment Application.

4.3.1 Clean-up from previous application (full-integration)

In order to comply with PCI DSS requirements, all files in the “datastore” folder of any previous installation of a Payment Application must be deleted.

The files in the “datastore” may include files that hold offline transaction information for delivery to the host. These files represent a financial value that will be lost if the files are deleted.

To avoid financial loss all offline transactions must be delivered to the host before the new application is installed. This is done by ensuring that the Payment Application is online and then activate the flush-operation of the API.

Any files that remain in the “datastore” folder after the flush-operation must be removed using a secure wipe-tool that erase and fill repeatedly with random data in accordance with industry accepted standards for secure deletion. Information on secure wipe tools for Windows and Linux systems are found in Appendix C. (PA-DSS 1.1.4 and PA-DSS 2.7).

If the operating system supports “restore points” (similar to functionality in Windows XP), these must be disabled during the clean-up procedure.

Failure to clean-up data from the previous application will render the new solution non-compliant with PCI DSS requirements.

Clean-up isn't required in embedded terminals where SW updates are handled by the supplied ftp-client and access to the file system is restricted.

4.4 Administrator Access (Configuration)

The application and its PA-DSS relevant configuration should be protected by user-accounts/passwords.

Application settings relevant to PA-DSS compliance are found in the "pcipadss.properties" file.

The specific requirements will depend on the type of installation and the operating system.

4.4.1 Full Integration

When installing for full-integration, it is required to establish separate user accounts for administration and daily operation (shop-assistant etc.).

Access to the "pcipadss.properties" file must require administrator privileges.

4.4.2 Embedded Integration (first generation platform)

Access to configuration files on the older embedded platform (see the following table for device characteristics) can only be achieved through physical access to the terminal.

Compliance with PCI DSS requirements can only be achieved by strictly following the procedure described in Appendix A.

4.4.3 Embedded Integration (second generation platform)

The embedded platform used in the PCI PED version of Flexi and in the UCM is delivered with a default administrative login through either console or via SSH.

To summarise the table below show the characteristics of various solutions and what is required to be compliant with PCI DSS requirements.

NOTICE: To achieve PCI DSS compliance the merchant/integrator must always replace default passwords with a password for the specific installation (PA-DSS 3.1-3.2).

The following PCI DSS requirements apply:

PCI DSS 2.3:

Encrypt all non-console administrative access. Use technologies such as SSH, VPN, or SSL/TLS for web-based management and other non-console administrative access.

Furthemore:

- Default settings and passwords should be changed and unique passwords used for each user.
- Where applicable (e.g., non dial up connections) connection to the remote site should be allowed from only specific IP addresses.
- Strong authentication or complex passwords should be used for each user according to PCI DSS requirements 8.1, 8.3 and 8.5.8 – 8.5.15.

PCI DSS 8.1 – Users should be assigned a unique ID.

PCI DSS 8.3 – A password and additional authentication item should be required to successfully authenticate.

PCI DSS 8.5.8.a – Generic user IDs and accounts should be disabled or removed. Shared user IDs for system administration activities and other critical functions should not be used. Shared and generic user IDs should not be used to administer devices in the environment.

PCI DSS 8.5.8.b – Use of shared access IDs, including their use for administrative purposes, is forbidden.

PCI DSS 8.5.8.c – Group and shared passwords are forbidden.

PCI DSS 8.5.8.b – Use of shared access IDs, including their use for administration purposes, is forbidden.

PCI DSS 8.5.8.c - Group and shared passwords are forbidden.

PCI DSS 8.5.9 – Configuration settings should be set to enforce users to change passwords at least every 90 days.

PCI DSS 8.5.10 - Configuration settings should be set to enforce user passwords to be at least seven characters long.

PCI DSS 8.5.11 - Configuration settings should be set to enforce user passwords to contain both numeric and alphabetic characters.

PCI DSS 8.5.12 - Configuration settings should be set to enforce new passwords to be different from the four previously used passwords.

PCI DSS 8.5.13 - Configuration settings should be set to enforce user accounts to be locked out after not more than six invalid logon attempts.

PCI DSS 8.5.14 - Configuration settings should be set to require a locked user account to remain locked for thirty minutes or until a system administrator resets the account.

PCI DSS 8.5.15 - Configuration settings should be set to require that system/session idle time out features are set to 15 minutes or less.

PCI DSS 4.1 - Encrypted data transmission protocols such as SSL/TLS or IPSEC should be used when accessing customer applications remotely.

PCI DSS 4.1.a – Encryption must be used wherever cardholder data is transmitted or received over open, public networks, furthermore

- Strong encryption should be used during data transmission.
- For SSL implementations, HTTPS should appear as a part of the browser Universal Record Locator (URL), and no cardholder data is required when HTTPS does not appear in the URL.
- Only trusted SSL/TLS keys/certificates are accepted.
- Proper encryption strength is implemented for the encryption methodology in use (based on vendor recommendations/best practices).

PCI DSS 4.1.1 – Integrators are advised that if a wireless network transmitting cardholder data is connected to the cardholder data environment, that industry best practices (for example, IEEE 802.11i) are used to implement strong encryption for authentication and transmission.

- For new wireless implementations, it is prohibited to implement WEP, as the deadline expired on March 31, 2009. For current wireless implementations, it is prohibited to use WEP after June 30, 2010.
- PCI DSS 8.5.13 - Configuration settings should be set to enforce user accounts to be locked out after not more than six invalid logon attempts.
- Where applicable (e.g., non dial up connections) a Virtual Private Network connection via a fire-wall must be established when accessing the application remotely.
- Customers are advised to enable logging of all remote access attempts on their systems.
- Access to customer passwords is restricted to authorised support personnel.
- User passwords must be established according to PCI DSS Requirements 8.1, 8.2, 8.4 and 8.5.

PCI DSS 8.1 – All users should be assigned a unique username.

PCI DSS 8.2 – The authentication methods should be consistent with the documentation.

PCI DSS 8.4 – Encrypt all passwords during transmission or storage on all system components.

PCI DSS 8.5.1 – Control the deletion and modification of user IDs, credentials and other identifier objects.

PCI DSS 8.5.2 – Verify user identity before performing password resets.

PCI DSS 8.5.3 – Set first-time passwords to a unique value for each user change immediately after each use.

PCI DSS 8.5.4 – Immediately revoke access for any terminated users.

PCI DSS 8.5.5 – Remove inactive user accounts at least every 90 days.

PCI DSS 8.5.6 – Enable accounts used by vendors for remote maintenance only during the time period needed.

PCI DSS 8.5.7 – Communicate password procedures for all users that have access to cardholder data.

PCI-DSS 8.5.8.a - Generic user IDs and accounts should be disabled or removed. Shared user IDs for system administration activities and other critical functions should not be used. Shared and generic user IDs should not be used to administer devices in the environment. The use of wireless is expressly prohibited.

PCI DSS 8.5.8.b – The use of shared access IDs, including their use for administration purposes, is forbidden.

PCI DSS 8.5.8.c - Group and shared passwords are forbidden.

PCI DSS 8.5.9 – Configuration settings should be set to enforce users to change passwords at least every 90 days.

PCI DSS 8.5.10 - Configuration settings should be set to enforce user passwords to be at least seven characters long.

PCI DSS 8.5.11 - Configuration settings should be set to enforce user passwords to contain both numeric and alphabetic characters.

PCI DSS 8.5.12 - Configuration settings should be set to enforce new passwords to be different from the four previously used passwords.

PCI DSS 8.5.13 - Configuration settings should be set to enforce user accounts to be locked out after not more than six invalid logon attempts.

PCI DSS 8.5.14 - Configuration settings should be set to require a locked user account to remain locked for thirty minutes or until a system administrator resets the account.

PCI DSS 8.5.15 - Configuration settings should be set to require that system/session idle time out features are set to 15 minutes or less.

PCI DSS 8.5.16- Authenticate all access to any database containing cardholder data. This includes access by applications, administrators, and all other users.

4.5 Verification of PCI DSS Settings

When the Payment Application is started it will check the settings of the “pcipadss.properties” file and verify whether any settings allow PAN data to be delivered by the Payment Application. The result of this verification will be indicated in the Initialisation Report.

This is sent on the API for the receipt printer. As integrators always have to support this interface it will be available to check the serial numbers of HW devices, version and configuration of the Payment Application etc.

A “mode” number in this report will indicate the availability of PAN information on the Payment API. The indication reflects the settings that the application found in the PCI DSS Configuration file.

- 0 Indicates that all PAN's passing the API are truncated
- 1 Indicates that full PAN is delivered when printing a signature receipt
- >4 Indicates that the Merchant Application does receive full PAN data

It is the intent that this mode can be checked by the acquirer as part of the integration testing.

4.6 Restrictions on storing Cardholder Data

The Payment Application stores the PAN securely as described in paragraph 3.1.

If cardholder data are collected outside the application by merchant/integrator, they must be purged after a customer defined retention period. (PA-DSS 2.1)

If PAN data are collected on databases etc., accounts that provide access to these data must use unique usernames/passwords (PA-DSS 3.2).

Cardholder data must never be stored on servers connected to the internet (PA-DSS 9.1).

The Payment Application doesn't prevent or interfere with any two-factor authentication the integrator/merchant may use for remote access to sensitive information stored outside the Payment Application (PA-DSS 11.2).

4.7 Network Connection

The equipment where the Payment Application is deployed is connected to a network. Sensitive data in the host communication is encrypted to fulfil PCI DSS requirements. This means that the communication itself can take place through an “open” network like the internet.

Fraud is still possible by gaining access to the payment platform through modification or replacement of SW components. As a consequence it is necessary to secure the local network as described below in order to comply with PCI DSS requirements.

The Payment Application is designed for use in a secure LAN environment and does not operate with or support any web servers. The application will support the use of a DMZ to separate it from any Internet systems.

It is recommended that local networks are protected by firewalls.

4.7.1 Wireless Networks

If the Payment Application is used in an integration that involves a wire-less network, it must be ensured that no PAN data is passed through the network in clear text and that the application and its configuration is protected (PA-DSS 6.1 – 6.2).

It will be necessary to protect the network with a fire-wall as described in PCI DSS requirement 1.2.3 (included below).

The wireless network must be set to use strong encryption for authentication and encryption. Please refer to PCI DSS 4.1.1 (included below).

PCI DSS 1.2.3:

Install perimeter firewalls. These firewalls must be configured to deny or control any traffic from the wireless environment into the cardholder data environment.

PCI DSS 2.1.1:

Customers are instructed to change default settings on wireless devices and ensure that all wireless networks implement strong encryption mechanisms including:

Encryption keys were changed from default at installation, and are changed anytime anyone with knowledge of the keys leaves the company or changes positions

Default SNMP community strings on wireless devices were changed

Default passwords/passphrases on access points were changed

Firmware on wireless devices is updated to support strong encryption for authentication and transmission over wireless networks (for example, WPA/WPA2)

Other security-related wireless vendor defaults, if applicable

PCI DSS 4.1.1:

Ensure wireless networks transmitting cardholder data or connected to the cardholder data environment, use industry best practices (for example, IEEE 802.11i) to implement strong encryption for authentication and transmission.

For new wireless implementations, it is prohibited to implement WEP after March 31, 2009.

For current wireless implementations, it is prohibited to use WEP after June 30, 2010.

4.8 SW Updates

Details on available PA-DSS approved builds, updates, ftp-site and web-site for checksums can be obtained by contacting the BBS Denmark support team.

4.8.1 Updating full integration deployments

Integrators that include the Payment Application as a fully integrated part of their solution will be notified when new versions of the Payment Application become available.

The new version can be downloaded from the Sagem Denmark download server.

Before deploying the new application, it is important to check that the MD5 checksum of the application jar-file corresponds with the checksum published on the Sagem Denmark website (PA-DSS 10.1).

4.8.2 Updating embedded deployments

Embedded terminals linked to the Sagem Denmark SW Download Server will automatically receive updated SW releases according to the agreement with Sagem Denmark.

The terminal contacts the ftp-server to check for available updates at configured intervals. If update files are available on the server they will be downloaded by the terminal and installed.

Configuration of the IP-address etc. for this connection is done through the administrative login. Information on how to configure the SW download (ftp-address, interval etc.) is included in the documentation.

4.9 Trouble-shooting

4.9.1 Debug-Logging

Log information from the Payment Application doesn't include PAN's or sensitive authentication data.

For performance reasons it is still advisable only to use minimum error logging unless specific debugging is in progress. The size of the cyclic log can be configured, but it is good practice to purge trace logs after use (PA-DSS 1.1.5).

4.9.2 Support Procedures

In order to comply with PCI PA-DSS Sagem Denmark has introduced new support procedures with the following restrictions:

- Sagem Denmark never access delivered applications or equipment through remote login (PA-DSS 11.3).
- Sagem Denmark can't accept full PAN information in neither written nor electronic form.

5 Configuration File Settings

A new configuration file named "pcipadss.properties" is included with the application to configure various PAN handling settings for integrations of the payment application.

5.1 General Security Settings

These settings must be configured as describe to achieve PCI DSS compliance.

Default settings mentioned indicate the function when a specific setting is not present in the configuration. Notice that the PCI PED compliant setting may not be the default setting.

5.1.1 Selection of Crypto Driver

This setting selects the crypto-engine used to perform the PAN encryptions needed when storing PAN's temporarily.

The setting is configured by Sagem Denmark in the package delivered. Normally this setting should not be changed.

Presently two versions exist: One standard java JCE and one for JBed solutions:

```
driver=dk.ascom.ps.services.pcipadss.StdJceCryptoEngine
```

or

```
driver=dk.ascom.ps.services.pcipadss.JbedCryptoEngine
```

The embedded PCI PED approved Flexi (plastic keypad with backlight), unattended payment solutions using the UCM platform and all full-integration solutions should use the standard crypto-engine (first example).

The second setting is used only for solutions using the JBed platform. These include embedded Flexi in the Visa PED approved design (rubber keypad).

NOTICE: In order to ensure PA-DSS compliance the Payment Application will provide truncated PAN's if this setting is not present or the configured engine isn't available. In some cases the Payment Application may refuse to perform online transactions if there no encryption engine is configured.

5.1.2 receipt.printFullPan

If set to true, the merchant receipts to be signed by the customer are printed with all PAN digits visible.

Cardholder receipts are still printed masked with X'es. As a minimum the PCI DSS required masking is performed. Some acquirers require even fewer digits visible.

This full PAN configuration is used when the merchant wants to use a signed receipt to retain the full PAN in offline transactions. Notice that if the transaction uses offline PIN verification no signature receipt is generated even for offline transactions.

Default value: false

To print full PAN on merchant receipt set:

```
receipt.printFullPan=true
```

5.2 API Settings for PAN Formats

This group of settings defines how to present PAN's when forwarded from the payment application.

5.2.1 Offered PAN formats

Setting	Handling
clear	The PAN is delivered in clear text.
full	Alias for clear
encrypted	Reserved for future use
truncated	The 6 first and 4 last digits is delivered, the digits in between are changed to X'es. This is the PAN truncation allowed by PCI DSS. <u>This is default for all API methods to be PCI PA-DSS compliant</u>
masked	alias for truncated
omitted	The API method is not called at all
hashed	The resulting PAN is hashed using a HASH-algorithm and a key. Presently only used by "e-Kvittering" implementation (see Appendix B).

5.2.2 PAN channels in API

The list of PAN channels is found in the table on the following page. The presentation of each channel may vary depending on the specific method of integration

List of PAN channels in the Payment Application API

Description	Method in Java API ("native")	Combridge integration (Note 1)	Serial Interface Integration (SIL)	PCI DSS friendly setting in pcipadss.properties	Comment
Card data from card swipe	cardData	CardData	CARD_DATA [61]	api.cardData=truncated	The truncation is special because this method handles full track not just PAN. If "truncated", the complete track is truncated. Track 1 and track 3 is delivered unchanged on the API
PAN received from an EMV chipcard	primaryAccountNumberEMV	CardPan	APPLICATION_PAN [84]	api.primaryAccountNumberEMV=omitted	
Transaction data delivered when the transaction is sent to the host	transactionData	TransactionData	TRANS_DATA [97]	api.transactionData=masked	
Journal data delivered when the delivery has been acknowledged by the host	addJournalData	JournalData	JOURNAL_DATA [64]	api.addJournalData=truncated	If no crypto-engine is available the PAN will be truncated even if the setting is set to "full"
Prompt shop-assistant to perform manual stoplist check	checkStopList	CheckStopListRequest	CHECK_STOP_LIST [46]	api.checkStopList=omitted	When "omitted" the payment application automatically answers the stop list question with the answer "Stop list not found"
Special feature in Danish application that offers reversal of previous transaction	checkReversal	CheckReversal		api.checkReversal=masked	Cannot be "omitted"

Note 1: The PA-DSS compliant version of the Payment Application requires a new version of Combridge where the data are embedded as parameters in the events.

6 Key Management

The Payment Application uses an internal key to protect PAN information.

The PANEncrypt key is stored in a separate keystore file. Normally all key happens automatically without any interference. The following information is given in case the keys have to be replaced.

The application will use one of three file-names for storing this PANEncrypt key depending on the platform:

- Either the key is stored in a Java Key Store (JKS) of type JCEKS (delivered by Sun). Keystore file named "ps.ks" is located in the "datastore" folder.
- Alternatively the key is stored in a Java Key Store (JKS) of type GNU Key Ring (part of the GNU Classpath Java implementation). Keystore file named "ps.gkr" is located in the "datastore" folder.
- In the JBed platform used for first generation embedded Flexi the keystore file is named "ps.ksj" is located in the "datastore" folder.

The keys is generated automatically by the Payment Application.

When the application is started it checks whether a key store file and a key-password file are present. If either file is missing a new set of files and a new key will be generated.

6.1.1 Updates of internal Keys

If more than 365 days have passed using the same key the application will generate a new key. All future card data will be encrypted using the new key.

This automatic update procedure will ensure that keys used for encryption of PAN is changed at least every 365 days (one year).

6.1.2 Replacement of internal Keys

If the internal keys have been compromised or if they have to be replaced for other reasons, the following procedure can be used:

- Delete the keystore file (see filename above)
- Restart application

When the application finds that the keystore file is missing it will automatically generate a new keystore and a new encryption key.

Appendix A: Configuration of First Generation Devices

The first generation of terminals delivered by Sagem Denmark A/S were based on the JBed platform.

JBed is a platform for running Java application in embedded systems.

Usually terminals are delivered with the required configuration already in place. But there may be rare occasions where the configuration has to be modified.

The JBed platform include a file system and features to access this file system by using a browser through the network connection.

It is unlikely that these features comply with the network security required to comply with PCI DSS. To make these older products compliant the new SW releases won't start the file access features.

In order to remain PCI DSS compliant when making changes to the configuration of these terminals the following procedure must be followed:

1. Disconnect the power from the Flexi terminal
2. Disconnect the network cable from the Flexi terminal
3. Connect a console (Hyper-terminal or similar) to the serial RS232 connector on the terminal (normally used for integration with an ECR).
4. Re-connect the power to the Flexi
5. Press "w" repeatedly on the connected console while the Payment Application is starting
6. The Payment Application will then start to communicate with the console to allow basic boot-time settings
7. One setting determines whether the file access is enabled – change this setting from 0 to 1
8. Continue the booting of the Payment Application
9. Connect a "cross-over" patch cable between the Flexi terminal an a PC
10. Start the browser on the PC in order to gain access to the file system in the terminal
11. Edit settings in configuration files as required (follow instructions on screen)
12. Disconnect network cable and serial cable from terminal
13. Reboot the terminal by disconnecting and reconnecting power
14. Connect the Flexi terminal to the network and the ECR
15. The Flexi is now operational with the new configuration

The file access feature is only activated once in the specific boot where the boot-setting was changed. The boot setting automatically resets to disable the feature at next boot.

Any attempt to configure the Flexi terminal without strict observation of the stated procedure will void the PCI DSS compliance of the installation.

Appendix B: PAN Hash for e-Kvittering

When the integration requires a unique identification of the cardholder, a hash value on the PAN can be generated.

Example of such applications are check-in/check-out parking applications and the electronic receipt from “e-Kvittering”.

The hash value can be delivered by the Payment API. Normally the Transaction Data message is configured to deliver the hash, but other parts of the API may also be configured to deliver the hash.

In order to configure TransactionData to deliver a hash (see: 5.2) include the following line in the pcipadss.properties file:

```
api.transactionData=hashed
```

The hash generation uses a SALT which is stored encrypted in two files. These files called “salt.ks” and “salt.txt” are obtained from BBS Denmark and placed in the “deploy” folder of the Payment Application.

Once the configuration is set and the files are in place, TransactionData will include a 20-byte hash value instead of the PAN.

For security reasons it may be required to update the “salt.txt” file. When a new value becomes available BBS Denmark will forward a replacement file.

Appendix C: Secure Wipe Tools

NOTICE: In order to avoid erasure of files containing offline financial transactions, it is important to perform a flush to ensure that transaction has been delivered to the acquirer before any remaining files are erased.

Deletion of sensitive data files.

To perform secure deletion of files, it is required to first overwrite the content of the file, with a random file pattern which will make it impossible for a file undelete program to read the original content of the file.

When a secure deletion of files is needed the following tools can be used:

Windows

SDelete

SDelete is a windows based command line tool which can be downloaded from www.sysinternals.com (see under file and disk utilities), and provides a way of writing random data to the file, by default only 1 time.

Usage:

1. Overwrite file with random data 25 times:
`sdelete -p 25 <filename>`

Linux

Shred

Shred is a command line tool which comes with most Linux distributions, and provides a way of writing random data to the file, by default 25 times. Shred will however not by default erase the file, unless it is specified.

Usage:

1. Overwrite file with random data command:
`shred <filename>`
2. Overwrite file with random data and delete file command:
`shred -u <filename>`